

6M-6

テキスト形式出力からの
論理図面自動生成アルゴリズム

若林 哲*, 石川 博之*, 三宅 祥二**, 平野 康彦**, 檀 良**

*法政大学大学院工学研究科電気工学専攻

**法政大学工学部電気工学科電気電子専攻

§ 1. はじめに

我々は、論理回路設計における回路自動修正機能 [論理チェッカー] [1]の研究を行っている。これは、論理回路設計者が設計した論理回路と期待値から、シミュレーション結果が一致し、正常動作する論理回路にゲート置き換えを行い、出力するものである。ここでは、その論理チェッカーの結果出力がテキスト形式で出力されたものをスキマティックな図面を自動的に生成し、画面等に出力するためのアルゴリズムについての研究である。これはゲート位置を自動的に配置し、結線を自動的に配線するものである。配置にはゲートの段位置のレベル付け[2][3]を行い、予め準備された配置位置に自動配置してゆく。配線は、迷路法[4]を用いて端子間の配線領域に自動配線してゆくものである。

§ 2. 論理チェッカーの概要

ここでは、論理チェッカーが論理ゲートの修正を施すためのアルゴリズムの概要について述べる。論理チェッカーとは、論理回路に対する出力の期待値を入力することにより、その期待値と論理シミュレーションの出力結果との一致、不一致にかかわらず、期待値信号にかかわるゲートをゲート単位で置き換え、シミュレーションを行い、その出力結果が期待値と一致した場合、そのゲートを置き換え可能ゲートとして出力するというものである。

そして、期待値を与えられている出力ピンに影響を与えるゲート全てを調べ終わると、結果として置き換え可能ゲートを、テキスト形式で表示するものである。

§ 3. アルゴリズムの概要

ここで報告するアルゴリズムは、全体図面をメッシュに切った升目を、ゲート配置領域と配線領域の2種類の領域に分けて考えてゆく。メッシュに切った升目を図3-1に示す。

3.1 配置アルゴリズム

まず最初に、論理ゲートをゲート配置領域に配置するため、論理ゲートにレベルを付けることから始まる。これは、入力端子から見てその

Automatic Routing Algorithm for Logic diagrams from text form outputs

Satoshi WAKABAYASHI*, Hiroyuki ISHIKAWA*, Shoji MIYAKE**, Yasuhiko HIRANO** and Ryo DANG**

*Graduate School of Hosei University,

**Hosei University

ゲートが何段目になるかをレベルで表す。図3-2で示すように、入力端子Aから見てゆくと、入力端子Aのレベルを1とし、Aに接続されている論理ゲートINVのレベルを2とする。その出力Dが接続しているAND2のレベルを3とする。次に、入力端子Bから見てゆくと、Bに接続するNOR2のレベルを2として、その出力Eが接続しているBUFFのレベルを3とする。さらに、その出力Fが接続しているAND2は、すでにレベルが3と付いているが、こちら側からみると4段目なので、新たにレベルを4に置き換える。同様に出力端子Gはレベルが4となっていたものも5となる。その結果、配置は図3-3に示すような配置となる。この方法は、入力端子から信号が伝達する方向へ調べてゆきレベル付けを行い、ゲートを配置してゆくため短時間で、少ないメモリで最適に近いゲート配置が実現できるという特徴を持っている。

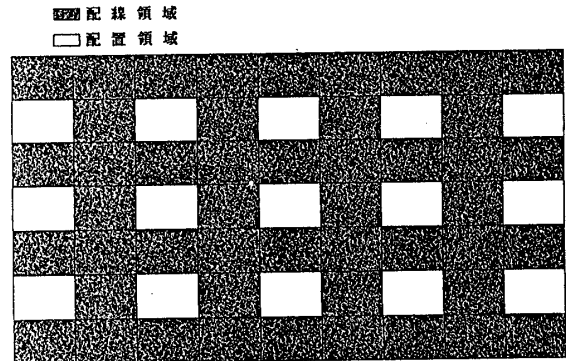


図3-1 配置配線領域メッシュ

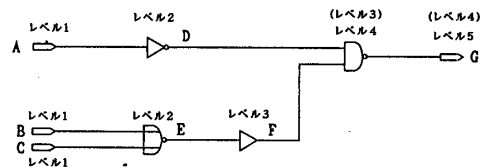


図3-2 ゲートのレベル付け

3.2 配線アルゴリズム

論理ゲートの配置が終わると、ゲート間の配線を始める。配線領域をさらに細かなメッシュに切り、配線禁止領域(配置領域)を除いて図3-4に示すように迷路法を用いて番号付けを行う。番号付けは、BUFFの出力端子FからNAND2の入力端子Hへ配線する場合にはF

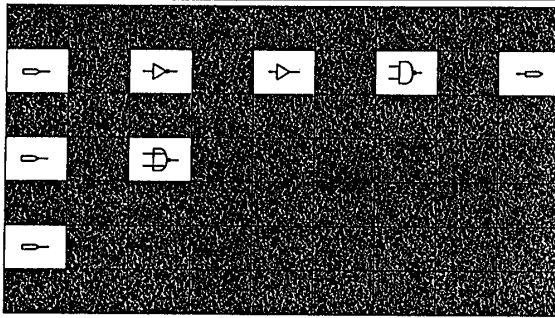


図 3 - 3 配置領域へのゲート配置

の端子を-3と置き、垂直水平方向に接しているメッシュを1、その次の垂直水平方向に接しているメッシュを2と置いてゆく。そして端子Fにたどり着くまで番号付けを行う。そして、端子Fから逆に、数値の1小さい方向へ水平方向に進んでゆく。もし、1だけ小さい値が存在しない場合は、垂直方向に同様に進んでゆく。そして、また1だけ小さい値が存在しなくなった場合に垂直方向へ進む。この繰り返しのよって、端子Hまでをたどってゆく。このとき、このアルゴリズムでは図面全体からF、H間のメッシュだけを切り出して配線を行っている。

このとき、水平方向に進むメッシュを(-10)、垂直方向を(-11)、水平から垂直への回転を上方向(-13)、下方向(-14)、垂直から水平への回転を上から左(-15)、下から左(-16)の様にメッシュに記録してゆく。既に配線がされている場合には水平(-10)、垂直(-11)以外は交差することができないため回転を余儀なくされる。水平(-10)、垂直(-11)の場合には、交差することが可能でメッシュは(-12)と交差を表す値に書き換えられる。また交点に対しては、表3-1に示す様にメッシュを書き換えてゆく。従来の手法では、回路図全体を見て配線を行っていたが、このアルゴリズムは図面全体のうちの配線する端子間の狭い範囲で処理を行い、全体図面に書き移すという手法を用いているため、理論上は配線ゲート数に制限がなく、メモリの制約はあるが、全体図面をファイルとして扱う事により相当大規模な図面生成を可能にしているといった特徴がある。

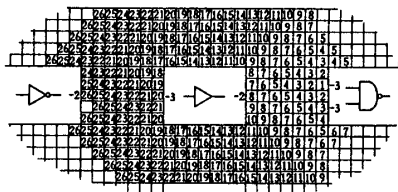


図 3 - 4 経路探索番号付け

経路進行方向	形状	番号	経路進行方向	形状	番号
横方向直進	—	-10	横方向・上分岐	┌	-17
縦方向直進		-11	横方向・下分岐	└	-18
交差	+	-12	縦方向・左分岐	┐	-19
左・上転回	┌	-13	縦方向・右分岐	┑	-20
左・下転回	└	-14	縦・横方向交差	+	-21
上・右転回	┐	-15			
下・右転回	┑	-16			

表 3 - 1 経路追跡用番号付け表

§ 4. 結果

以上のアルゴリズムをもとにパソコン上でシミュレーションを行った。その結果、図3-2の回路をもとに配置及び配線シミュレーションを行った結果を図4に示す。

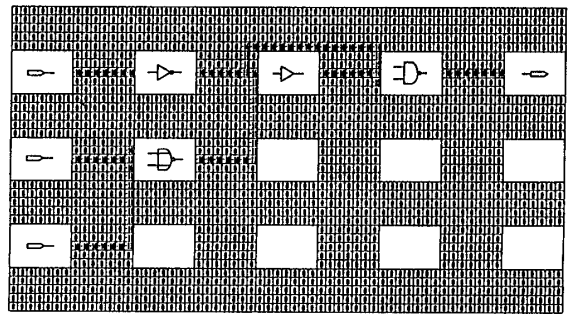


図 4 配線番号付け結果

§ 5. おわりに

テキスト形式出力から自動的に論理回路図をスキーマティックな形式で生成するためのアルゴリズムを考案し、我々の開発している「論理チェッカー」に組み込んだ。その結果まだ多少の問題点が挙げられる。

- (1)配置領域を上から詰めてゆくため配線の時に直進ではなく遠回りする場合がある。
 - (2)配置領域でのゲート配置位置が固定のため、直線配線が行えない場合がある。
- 以上の点を考慮し、今後のアルゴリズムの見直しを行ってゆきたい。

参考文献

[1]若林他, "論理チェッカー:論理回路設計への異常回路修正機能の付加", 信学技報VLD88-114, 1989
 [2]真鍋他, "機能論理設計のための図面生成用自動配線アルゴリズム", 第33回情処全大2R-5, 1986
 [3]西尾他, "自動論理合成システムLUNAの適用・評価-合成回路図面生成-", 第34回情処全大4F-8, 1987
 [4]C. Y. Lee, "An Algorithm for Path Connections and its Applications", IRE on Trans. on EC, vol. EC-10, no. 3, p. 346-365, 1961